Detection and Classification of Unauthorized Code as a Sensor for Trusted Insider Hostile Activity

by
Rusty Miller (rusty@eor.us)
EOR, Inc. www.eor.us

Abstract

A relatively new software technology, developed for use in a commercial-off-the-shelf (COTS) security product, has enormous potential for detecting a trusted insider's use of unauthorized software. With additional engineering and integration efforts, this COTS program could not only identify that the code was unauthorized, but it could also compare the code with a database of exploit code technologies known to compromise system security or software tools that support espionage operations. This capability would enable investigators to identify the computer user responsible for launching the hostile code, and also permit the investigators to open an investigation to determine the insider's purpose, if he is working for a foreign intelligence service (FIS), and what kind of information he may be trying to obtain.

Background

In February 2003, The President's Critical Infrastructure Protection Board released the "National Strategy to Secure Cyberspace". Amongst the document's many recommendations, there was a recommendation to:

> *"…ensure a strong counterintelligence posture to counter cyber-based intelligence collection against the United States government, and commercial and educational organizations. This effort must include a deeper understanding of the capability and intent of our adversaries to use cyberspace as a means for espionage.* [1] *"*

This recommendation recognizes the fact that FIS have always focused on gaining access to the information that their sponsor's require, and that the majority of information that they require probably exists somewhere in a digital form.

Clandestine espionage operations by definition strive to remain unnoticed. The widespread fielding of intrusion detection systems across government and DOD systems means that there is considerable risk to any FIS espionage operation that relies on obtaining access to information by penetrating systems from the outside-in. A more surreptitious approach to information gathering would be to begin where the information resides, on the inside. Individuals or computer programs operating inside of network-perimeter security often are given little scrutiny and are treated as trustworthy.

[1] National Strategy to Secure Cyberspace, http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf

The threat of malicious activity from a trusted insider with cyber access is well known. A 1998 article from the DOD Security Institute referred to the "Indispensable Role of the Insider"[2]

> *"It is important to note that the efforts of "outside" groups (including foreign interests) could be aided significantly by the assistance of parties within the organization with access to, and knowledge of, critical information systems. For certain secure, self-contained systems, the insider's access will prove indispensable. Whether the insider is recruited directly, indirectly (e.g. "false flag" recruitment), coerced through blackmail, or through "social engineering" is manipulated while unaware that he is providing assistance to an adversary, his collaboration is a tremendous force multiplier. The potential damage an insider can now commit has also been increased within the last decade by two related trends in information systems -- consolidation and, for all intents and purposes, the elimination of the need-to-know principle. These changes, designed to improve information sharing, have removed obstacles to hostile collection. The hostile, sophisticated information technology professional now has many more opportunities to enter and damage larger systems."*

FIS-recruited hostile insiders could be provided with a variety of tools to further their espionage aims. One possible scenario is one where FIS provides the hostile insider with tools to help the insider gain access to the information that the FIS is seeking. Such tools would be used to surreptitiously exploit system security and help the hostile insider gain unauthorized access, or access in excess of his authority. Another possible scenario is one where the hostile insider may already have access to the information that the FIS requires, and is instead provided with tools to support the clandestine communications requirements of the espionage operations.

Yet another hostile code scenario is one where FIS-recruited hostile insiders employ "Targeted Trojans", similar to the one recently discovered in Israel[3]. Or if not directly emplaced by a hostile insider, such programs might only require that an unwitting trusted insider visit a particular web site, and the Trojan program would be introduced to the user's system via "a drive-by download"[4]. The trusted insider might not even be aware that he was selected based on his access, and directed to the web site for the purpose of compromising his system's security.

For the purposes of this paper, we will refer to any software tools that further FIS goals as "hostile code" or "hostile executables".

---

[2] Security Awareness Bulletin No. 2-98, http://www.dss.mil/search-dir/training/csg/security/Treason/Infosys.htm
[3] Israel espionage case points to new Net threat, http://www.msnbc.msn.com/id/8145520/page/2/
[4] Definition of "Drive-by download", http://en.wikipedia.org/wiki/Drive-by_download

Discussion

There is a need in the counterintelligence and law enforcement communities for a means to detect trojanized executables and other forms of malicious code. Investigators cannot respond to FIS espionage-sponsored system penetrations via Targeted Trojans if they cannot detect them. A June 2002 National Needs Assessment stated:

> *"Investigators indicated that a law-enforcement-specific database to search for Trojans, root kits, and other known attack tools would be a valuable data resource. By linking to a trusted data source that is continually updated, investigators would be afforded relevant and timely attack analysis capability."*[5]

Espionage operations strive to remain undetected, and if detected, to remain unrecognized as espionage operations. For this reason it is more likely that a FIS-recruited, hostile insider would be provided with readily available exploit tools or clandestine communications technologies that are sufficient to support FIS operational goals, but that do not represent FIS state-of-the-art technology. This would allow a FIS-recruited, hostile insider to plausibly deny his espionage activities, and claim that he was merely a curious security "experimenter" with no specific intent, or give some other incriminating, but non-espionage explanation for his activities. On the other hand, if he is detected using FIS state-of-the-art exploit technologies, there are few plausible explanations other than espionage.

If it is true that FIS would rather use tools that do not reveal advanced capabilities, then the process proposed in this paper, that of identifying unauthorized code in real-time and matching it against a database of known hostile code, may have considerable utility in detecting espionage by trusted insiders.

If unauthorized code is detected and it does not match anything in the database of known hostile code, then investigators/analysts can analyze the unauthorized code to determine its purpose. In this manner analysts may identify a new clandestine communications tool, a Targeted Trojan, or some other tool used to further FIS goals.

Comparison of Security Models

Currently, most malicious code detection technologies employ a "permit all, except" model, where all code is allowed to execute unless it is on a list (a signature file) of known, malicious code. If a signature-based malicious code detection technology does not have an original example of malicious code, it cannot derive a signature. Unknown, or hitherto unseen malicious code can defeat a signature-based approach.

---

[5] Law Enforcement Tools And Technologies for Investigating Cyber Attacks, A National Needs Assessment, http://www.ists.dartmouth.edu/TAG/needs/ISTS_NA.pdf

The technology proposed in this paper is based on a "deny all, except" model, where all code is denied execution unless it is on a list of authorized code. Such an approach does not depend on having examples of all forms of malicious code, and is capable of detecting unknown, and possibly malicious code because the suspect code is not on the list of authorized code.

Clearly, the "deny all, except" model is more effective than the "permit all, except" model.

Authenticated Execution

The COTS program under consideration in this paper uses a technique called "authenticated execution". The idea of authenticated execution is that before a program loads into system memory, a cryptographic hash of the program is generated and is compared against a database of cryptographic hashes of all authorized executables for the system. If the program's hash matches an authorized hash, the executable is permitted to load into memory. If the program's hash does not match that of any authorized executable, the program is not allowed to load into memory and execute. Also, when an unauthorized program is detected, both its cryptographic hash and a notification of a security event are sent to the COTS program's security management console.

The fact that the cryptographic hash of the unauthorized executable is sent to the COTS program's security management console is of particular significance. It means that a database of the cryptographic hashes of known, hostile executables can be prepared and the cryptographic hash of the unauthorized program can be compared against the database. If there is a match, then investigators will know that someone (or some process) tried to execute hostile code on a specific system. This knowledge would allow investigators to open an investigation to determine the circumstances surrounding this suspicious event. In effect, the COTS program (suitably enhanced) becomes a distributed sensor for hostile activity by trusted insiders.

Known Hostile Code Database

A key element in the concept proposed in this paper is the creation of the database containing the cryptographic hashes of known hostile code. Examples of known hostile code will fall into two categories; unclassified system exploit or communications tools commonly found on the Internet, and classified FIS system exploit or communications tools that have been obtained by US intelligence agencies.

The process of populating the database will require collecting a large set (as large as possible) of hostile code tools obtained from both the Internet, and from US intelligence agencies. Every tool will need to be run through a hashing function (SHA1 is the default function for the COTS software), and the resulting hashes will be entered into the database.

Note: Because the hostile code database will contain only cryptographic hashes, it will be completely unclassified. Further, US intelligence agencies that may be reluctant to provide actual classified FIS tools that have been collected during US intelligence operations, may

instead be willing run the FIS tools through the SHA1 hashing function themselves and contribute just the hashes to the hostile code database.

There is an additional benefit of the fact that the hostile code database will contain only unclassified cryptographic hashes. Because the hostile code database will be unclassified, it can be on-line and available for real-time hash matching. This means that the moment a hostile insider tries to execute any type of unauthorized code, the cryptographic hash of that code can be matched against the hashes in the hostile code database.

Counterespionage Benefits

If adopted, the techniques proposed in this paper would enable investigators/analysts to:

- Detect & observe the activities of a hostile insider
- Enable a centrally managed, espionage detection, analysis and exploitation center
- Determine the extent and nature of hostile insider activities
- Analyze the technical tradecraft used in cyber-espionage
- If the counterespionage capabilities are intended to remain undisclosed, the COTS program's information assurance benefits provide ready-made cover for the presence of the COTS software
- If the counterespionage capabilities are publicized, the presence of the COTS program serves as a deterrent to espionage

Information Assurance Benefits

If adopted, the COTS software described in this paper could:

- Stop all unauthorized code – even unknown attacks
- Provide enterprise situational awareness of unauthorized executables
- Permit rapid analysis & response to threats
- Enable centrally managed host based sensors for unauthorized code
- Reduces virus entry vectors
- Provide enhanced system security without depending on training or policy adherence
- Control configuration management & maintain accreditations (unauthorized installation programs won't run)

Conclusion

There is a demonstrated need for a capability to detect trusted insiders when they attempt to run unauthorized, possibly hostile software tools. The COTS software described in this paper, suitably enhanced with a database of known hostile code, could provide such a capability.

In order to test and refine the hostile activity detection capabilities described above, recommend that a government sponsor field a small pilot project to measure the effectiveness of the technical

investigation concepts discussed in this paper. Part of the project's evaluation criteria should include the controlled introduction of known hostile code to test the core concept of the project.

The project sponsor should also consider deploying the project technologies to several different organizations to test its capabilities in diverse technical environments and to increase the chance of detecting real hostile insider activity.

The scope and specific capabilities of the proposed pilot program can be further defined in a detailed statement of work, to be provided upon request by a government sponsor.